Notice of Allowability	Application No.	Applicant(s)	
	09/901,254	SOLLICH, PETER F.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *11/03/2004*.

2. ☒ The allowed claim(s) is/are *1-42*.

3. ☒ The drawings filed on *09 July 2001* are accepted by the Examiner.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date *2005/02/04* .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/03/2004.

As indicated in Applicant's response, claims 3, 16 have been amended and claims 41-42 added. Claims 1-42 are pending in the office action.

EXAMINER'S AMENDMENT

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Attorney Himanshu S. Amin, Reg # 40,894 on 2/3/2005.

The application has been amended as follows:

In the CLAIMS:

Claim 1:

A computer implemented system for facilitating an interface dispatch, comprising:
a pre-execution engine ~~adapted to~~ that loads source code and allocates a block of memory in the form of a vector for creating an interface map that includes a plurality of slots for referencing interface virtual tables that correspond to an implementation of an interface in a class type; and
an interface index component ~~adapted to~~ that assigns index numbers to interfaces as the interfaces are loaded by the pre-execution engine, the pre-execution engine ~~being adapted to~~ determines a row structure ~~for a class type for a plurality of class types~~ based on ~~a class type and~~ interfaces implemented by the class type using the indices assigned to the interfaces, the pre-execution engine associating indices with empty slots in the interface map based on the

configuration of the row structure and storing references to the interface virtual tables in the empty slots when enough empty slots are found for the respective row structure.

Claim 2:

The system of claim 1, the pre-execution engine ~~being adapted to~~ stores row structures utilizing a comb-vector technique.

Claim 4:

The system of claim 1, the pre-execution engine ~~being adapted to~~ assigns a row start location upon finding empty slots for each interface index entry, the row start location providing a start location for the row structure and the interface index numbers providing offsets from the row start location.

Claim 14:

The system of claim 1, the pre-execution engine ~~being adapted to~~ dynamically creates additional interface maps as the interface maps are filled.

Claim 15:

The system of claim 1, the pre-execution engine ~~being adapted to~~ creates a special map for all COM classes and COM interfaces.

Claim 16:

The system of claim 1, further comprising an execution engine ~~adapted to~~ receives a method call of an interface for a specific class type and access the interface virtual table corresponding to the interface and class type utilizing the interface map.

Claim 18:

A computer implemented method of creating a reference map to interface virtual tables, comprising:

allocating a vector in memory of a predetermined size having a plurality of slots for creating an interface map for storing references to interface virtual tables;

loading source code having a plurality of classes and plurality of interfaces;

assigning indices to the plurality of interfaces;

determining a row structure for a class type and interfaces implemented by the class type utilizing the indices; ~~and~~

storing references to interface virtual tables corresponding to the row structure in the plurality of slots utilizing a comb-vector technique; and

associating indices with empty slots in the interface map based on the configuration of the row structure and storing references to the interface virtual tables in the empty slots when enough empty slots are found for the respective row structure.

Claim 19:

The method of claim 18, further comprising repeating the ~~steps~~ acts of determining a row structure and storing references for each class type that implements interfaces in the source code.

Claim 28:

A computer-readable medium having computer executable components comprising:

a pre-execution engine component ~~adapted to~~ that loads source code having a plurality of classes implementing interfaces, the pre-execution engine assigning indices to the interfaces as the interfaces are loaded;

an interface map component that includes a plurality of slots for referencing interface virtual tables that correspond to an implementation of an interface in a class type, the pre-execution engine ~~being adapted to~~ determines a row structure for a class type based on ~~a class type and~~ interfaces implemented by the class type using the indices assigned to the interfaces, the pre-execution engine associating indices with empty slots in the interface map based on the configuration of the row structure and storing references to the interface virtual tables in the empty slots when enough empty slots are found for a respective row structure.

Claim 29:

The computer readable medium of claim 28, further comprising an execution engine component ~~adapted to~~ that receives and executes code and accesses the interface virtual table corresponding to the interface and class type utilizing the interface map in response to a method call for a specific interface type implemented in a class instance of a specific class type.

Claim 31:

The computer readable medium of claim 28, the pre-execution engine ~~being adapted to~~ stores row structures utilizing a comb-vector technique comprising conceptually sliding row structures corresponding to class types within the interface map until each interface index entry hits an empty slot.

Claim 32:

The computer readable medium of claim 28, the pre-execution engine component ~~being adapted to~~ assigns a row start location upon finding empty slots for each interface index entry, the row start location providing a start location for the row structure and the interface index numbers providing offsets from the row start location, the row start location being stored in a method table corresponding to the class type associated with the row structure.

Claim 34:

The computer readable medium of claim 28, the pre-execution engine component ~~being adapted to~~ creates the interface map.

Claim 35:

The computer readable medium of claim 34, the pre-execution engine component ~~being adapted to~~ dynamically creates additional interface maps as the interface maps are filled.

Claim 36:

The computer readable medium of claim 34, the pre-execution engine component ~~being adapted to~~ creates a special map for all COM classes and COM interfaces.

Claim 38:

A computer implemented system for facilitating an interface dispatch by providing an interface map containing references to interface virtual tables, comprising:

means for allocating a block of memory for creating an interface map with a plurality of slots;

means for loading source code having a plurality of class types implementing interfaces;

means for assigning indices to interfaces as the interfaces are loaded;

means for determining a row structure based on the class type and the interfaces implemented by the class type employing the indices; ~~and~~

means for storing references to interface virtual tables based on the row structure of each class type using a comb-vector technique; and

means for associating indices with empty slots in the interface map based on the configuration of the row structure and storing references to the interface virtual tables in the empty slots when enough empty slots are found for the respective row structure.

Claim 40:

The system of claim 38, further comprising means for accessing the references in the slots, the means for accessing the references using a row start location corresponding to a class type and adding the index number associated with the desired interface to the row start number to access the reference to the interface virtual table.

Claim 41:

~~An~~ A computer implemented interface dispatch system, comprising:

a pre-execution engine that loads code and allocates memory for an interface map in the form of a one-dimensional array that includes a plurality of memory slots that reference interface virtual tables that provide references to implementations of one or more class interfaces; and

an interface index component to assign index numbers to interfaces as they are loaded by the pre-execution engine, the pre-execution engine determines a row structure defining spatial relations amongst interfaces of the same class based on the ~~class,~~ interfaces implemented by the class and the indices assigned to the interfaces, associates indices with empty slots in the interface map based on the row structure, and stores references to interface virtual tables in the empty slots if the map can accommodate the row structure, otherwise a new interface virtual table is created.

Claim 42:

A computer implemented method of accessing interface implementation methods comprising:

allocating an interface map as a one-dimensional array that includes a plurality of memory slots that reference interface virtual tables used to provide references to implementations of one or more class interfaces; associating indices with empty slots in the interface map based on a row structure which defines spatial relations amongst a plurality of interfaces of the same class; and storing references to the interface virtual table in the empty slots when enough empty slots are found for a respective row structure; said storing comprising:

receiving a method call associated with a particular interface;

determining the class that implements the interface;

retrieving an interface map row start value from a method table associated with the class that implements the interface, the row start identifying the first slot of a row structure~~,~~ ~~which defines spatial storage relations amongst a plurality interfaces of the same class by indicia, for the particular class~~;

retrieving an index offset associated with the interface;

adding the offset to the row start value to locate the slot in the row structure

associated with the particular interface;

retrieving the address for the appropriate interface virtual table from the located slot; and

locating and executing the received method via the interface virtual table.

EXAMINER'S STATEMENT OF REASONS FOR ALLOWANCE

3. Claims 1-42 are allowed.

The following is an examiner's statement of reasons for allowance.

The prior art of record, taken alone or in combination fails to teach or suggest the following claimed features:

A method for facilitating an interface dispatch comprising an pre-execution engine to load source code and (i) allocate vector in memory to create an interface map including a plurality of slots for referencing virtual tables that correspond to an implementation of an interface of a class type; (ii) an index component to assign index numbers to interfaces as the interfaces are loaded by the pre-execution engine, the pre-execution engine determining a row based on interfaces implemented by the class type using the indices, and associating indices with empty slots in the interface map based on the configuration of the row structure and storing references to the interface virtual tables in the empty slots when enough empty slots are found for the respective row structure – as recited in claims 1, 18, 28, 38 and 42; or (iii) if not, creating new interface virtual table as in claim 41.

Bottomley, USPN: 20040015912, in a method to load class and corresponding interfaces implemented by the class, discloses a system interface table listing all the interfaces needed by the code, such interfaces referring to entries thereof into a master interface table (MITABLE),

such entries filling originally null spaces a sliding manner as more interface entries are created, each of such space representing a reference to a virtual table (IVTABLE) storing all the methods for the interface being stored in the master table. The empty space being filled here are not index created for interfaces that are loaded based on a particular class type; and the filling of empty slots is not based on configuration of row structure defined by the index component as in (ii) and not solely based on whether the slots being found to accommodate the row structure provide enough space therefor as in (ii) and if not, to create new interface virtual table as in (iii).

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
February 3, 2005

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100